## linia pozioma

Hackathon

March 2020

# Explanation of the assignment

The hackathon is being organized to find new and creative solutions to a lot of manual work at DRBC. To solve this problem, automatic tagging based on NLP is being considered. Participating companies can freely choose which tool they use. Below are some examples of how this assignment could be carried out. As a first step one of these pre-trained word vector models could be used:

<https://fasttext.cc/docs/en/crawl-vectors.html>

<http://vectors.nlpl.eu/repository/>

Via these models words and terms can be coded in a "meaning vector". The intention is to:

1. give different tag words.

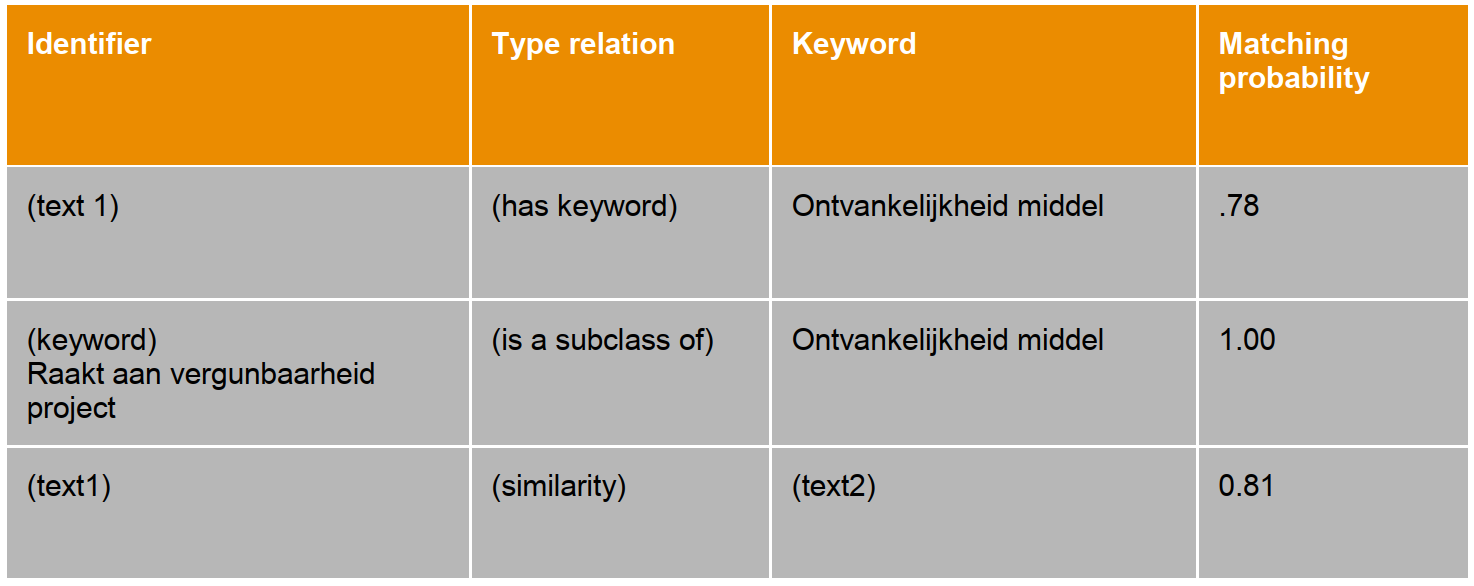
2. match with each other.

You can do this for example through getting rid of frequent words. After this the average word2Vec vector can be calculated for the remaining words (we recommend cutting the text into pieces, since different paragraphs may contain different themes). An example of this is:

<http://www.rpubs.com/mukul13/rword2vec>

<https://github.com/mukul13/rword2vec>

These average vectors can then be compared with the vectors of the tag words for their similarity. This can for example be determined by cosine similarity. If the matching exceeds a certain score (eg 0.85), you can assume that the text and the tag word belong together. The result will therefore be a series of triples or quadruples:



Other methods that can additionally work are classical regex matching, bag-of-words, possibly transfer learning of the existing word2vec models direction, more legal language, LDA or other NLP techniques.

# The supplied data & evaluation of the results:

The DBRC already has various data sources for achieving better access to the case law. More specifically, the information from these documents will be shared:

● 1000 PDFs of the Judgments, which you can also find public here:

<https://www.dbrc.be/rechtspraak>

● One digitalization glossary: main labels with several sub-labels underneath.

● Overview of EVR documents: These contain the type of judgment per file and manually

added tag words

Part of the texts that are manually tagged (EVR documents) will be used for the evaluation of the results in terms of Accuracy, recall and precision. <https://en.wikipedia.org/wiki/Precision_and_recall>

# Match what?

1) Matching must be done with the digitization glossary

2) Matching must be done with the EVR glossary

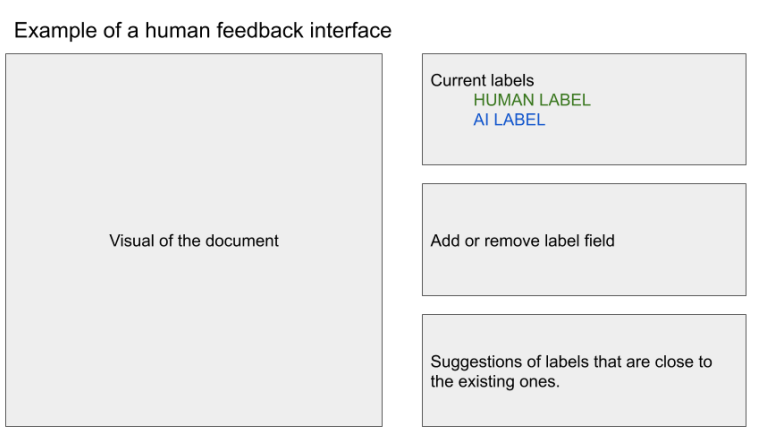
3) A matching between judgments themselves

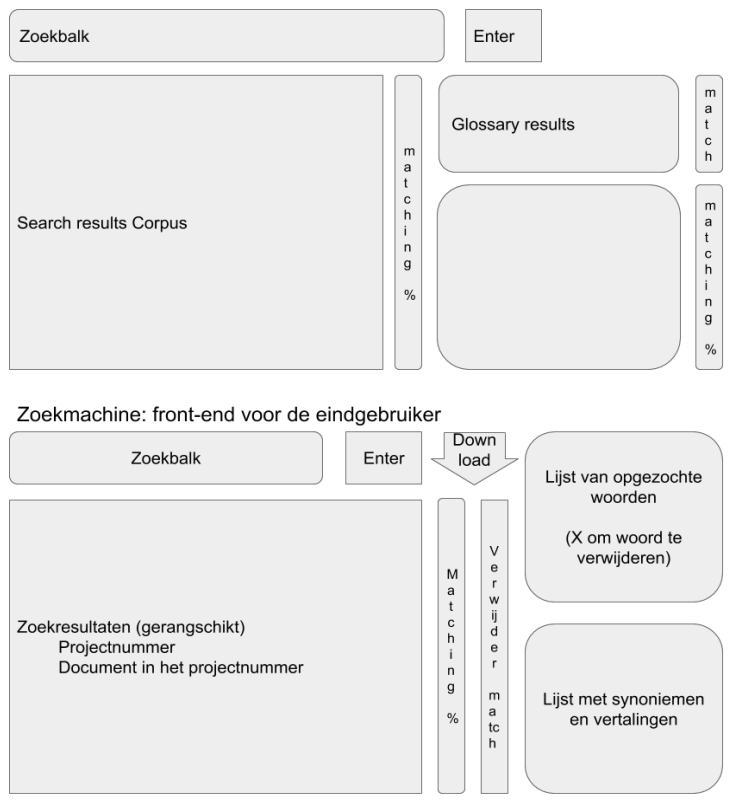
Additionally, a clustering can be done based on the EVR glossary (similar words or sentences with the same meaning can be grouped). And in addition, it can be interesting to do during the hackathon (in addition to the aforementioned matching) also to develop functionality to search for new topics in the judgments. This section is optional.

# Front-end and visualization

In addition to the back-end engine, we also want to see a demo of the front-end. From the collected feedback from users, there is already an existing example found to be useful: <http://juridict.raadvst-consetat.be/index.php?lang=en>

A similar look-up structure therefore also seems a useful way of working. There are also ways of visualization such as the example diagrams below:





# Delivery of results

The following documents must be submitted for the final evaluation:

● A data file with the links between:

○ Documents and their keywords

○ Documents and their mutual relationship

○ Keywords and their hierarchy.

● A separate data file in the same format as above for the "test set": this

will be used to evaluate the model. This file must be in .txt, .csv or .xlsx

to be supplied.

● A demonstration of the front end on Friday, March 13

All results achieved are displayed in an open way and shared with all candidates at the hackathon. The data files may be forwarded to christophe.cop@pwc.com